

修 士 論 文 概 要 書

Summary of Master's Thesis

Date of submission: 1 / 31 / 2012 (MM/DD/YYYY)

専攻名（専門分野） Department	情報理工学専攻	氏 名 Name	市村 勇貴	指 導 教 員 Advisor	甲藤 二郎 印 Seal
研究指導名 Research guidance	画像情報研究	学籍番号 Student ID number	CD 5110B016-7		
研究題目 Title	無線マルチホップアドホックネットワークに適した低遅延型 TCP 輻輳制御方式				

1 はじめに

近年の無線技術の発達により、無線 LAN やモバイルアドホックネットワークなどが研究開発されてきている。将来のユビキタス社会に向けて、無線 LAN のようなアクセスポイントやネットワークインフラを用いず、移動端末のみで自律分散的にネットワークを構築するアドホックネットワークが注目されている。アドホックネットワークの適用例として、会議場、災害現場、軍事利用などでの活用が期待されている。特にマルチホップアドホックネットワークは、有線ネットワークと異なり、全てのノードが無線チャネルを共有しているため、ホップの増加やウィンドウサイズの増加に伴い、ノード間においてデータ送信機会が減る。その結果、性能が低下するという問題がある。また、従来の TCP は有線向けに設計されているためにウィンドウサイズを積極的に増加させる。そのため、無線マルチホップアドホックネットワークでは MAC 層でのコンテンツionやパケット衝突、再送タイムアウトなどを引き起こし、性能低下に繋がる。そこで、TCP Vegas[1]を基にしたウィンドウサイズの増加幅・増加回数を抑制する輻輳制御方式[2]が提案されている。この方式は、ノード配置がシンプルな環境では効果的に作用する。しかし、現実的で複雑な環境では効果的に作用しないことがあるため、本研究ではパケットロス後の対応を改良することで、改善を目指す。

2 従来研究

2.1 TCP Vegas [1]

TCP Vegas は、送信パケットの RTT を観測し、その変動をウィンドウサイズに利用する delay-based 手法である。RTT が大きくなれば、輻輳が発生していると判断してウィンドウサイズを小さくし、RTT が小さければ、逆にウィンドウサイズを増加させる。

式(1.1)に示すルータのキュー内の滞留パケット数(Δ)の推定値を用いることで制御する。

$$\Delta = \frac{cwnd}{RTT_{\min}} - \frac{cwnd}{RTT} \quad (1.1)$$

2.2 TCP NewVegas [2]

[2]では、ウィンドウ増加回数の抑制とウィンドウ増加幅の抑制を TCP Vegas に用いる手法(以下 NewVegas)が提案されている。この手法はウィンドウを小さく抑制することにより、ネットワーク負荷を低減させ、スループットを向上させることを目標としている。増加前のウィンドウサイズを $cwnd$ 、増加後のウィンドウサイズを $cwnd_{new}$ 、スロースタートフェーズを、以下の式(1.2)のように表す。 p 、 γ は定数とする。

連続して($\Delta < \gamma$)フェーズに入った回数を $succ$ とする。増加幅制御パラメータを $count$ 、 $succ$ 最大値の閾値を N (例: $N=10$)と設定する。

$succ$ が N より大きくなるまで $cwnd$ を維持し、 N を超えたら $1/(2 \times count)$ 分だけウィンドウサイズを増加させ、 $succ$ を初期値の 0 に戻し、 $count$ を 1 増加させる。これにより、ウィンドウ増加幅は Vegas では常に 1 であるのに対し、NewVegas では徐々に小さくなっていく。

$$cwnd_{new} = \begin{cases} cwnd + 1/(2 \times count) & (\Delta < \gamma \ \& \ succ > N) \\ cwnd & (\Delta < \gamma \ \& \ succ \leq N) \\ cwnd \times (1 - p) & (\Delta \geq \gamma) \end{cases} \quad (1.2)$$

輻輳回避フェーズにおいても、スロースタートフェーズと同様にウィンドウサイズの増加幅を Vegas より小さく保つ。輻輳回避フェーズを以下の式(1.3)のように表す。連続して($\Delta < \alpha$)のフェーズに入った回数を $succ2$ とする。増加幅制御パラメータを $count2$ 、 $succ2$ の最大値の閾値を N (例: $N=10$)と設定する。

$$cwnd_{new} = \begin{cases} cwnd + \frac{1}{cwnd \times 2 \times count2} & (\Delta < \alpha \ \& \ succ2 > N) \\ cwnd & (\Delta < \alpha \ \& \ succ2 \leq N) \\ cwnd & (\alpha \leq \Delta \leq \beta) \\ cwnd - \frac{1}{cwnd} & (\Delta > \beta) \end{cases} \quad (1.3)$$

3 提案手法

TCP がパケットロスを検知する方法として、1)重複 ACK を 3 回受信する、2)再送タイムアウト、の 2 種類ある。提案手法はこの 2 種類に対して、それぞれ以下のような改良を行う。

3.1 重複 ACK

従来の NewVegas は、重複 ACK に対してウィンドウ半減させる。提案手法として、TCP Westwood[3]を参考に、RTT を用いた動的なウィンドウ減少に変更する。

$$cwnd = cwnd \times \frac{1}{2} \rightarrow cwnd = cwnd \times \max\left(\frac{RTT_{\min}}{RTT}, \frac{1}{2}\right)$$

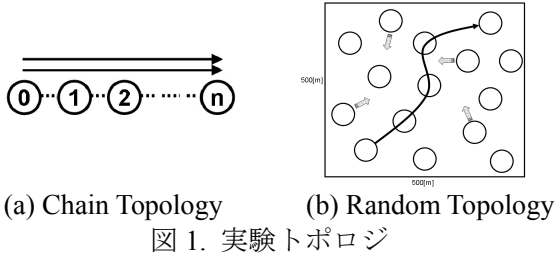
(従来手法) (提案手法)

3.2 再送タイムアウト

従来の NewVegas は、再送タイムアウトに対してウィンドウサイズを 1 に戻した後、スロースタートからやり直す。提案手法として、LDA-RTO[4]を参考に、再送タイムアウトの原因を判別し、原因が非輻輳によるものならば、再送タイムアウト時の状態(ウィンドウサイズやスロースタート閾値)に戻す。

4 シミュレーション

ns-2を用いてシミュレーション実験を行った。図1の(a)Chainトポロジ、(b)Randomトポロジを用いて、従来手法(Reno, Vegas, CUBIC, Compound, NewVegas)との比較評価を行った。帯域幅は2[Mbps]、ルーティングプロトコルにAODV、MAC層に802.11、キューサイズを50[pkt]、パケットサイズを1000[byte]、パケットエラー率を1[%]、通信可能範囲150[m]、キャリアセンス範囲を250[m]、シミュレーション時間300[sec]とする。



4.1 Chain トポロジ

まず、シンプルなChainトポロジにて実験を行う。フロー数を1,4[flow]、ホップ数を1~10[hop]、ノード間距離を100[m]、とする。各方式のスループットと平均RTTをそれぞれ図2と図3に示す。

1フローの場合、図2(a)と図3(a)より、スループットと平均RTTともに、提案手法はNewVegasとほぼ変わらない結果である。1フローの場合は、他フローとの競合がないため、送信権の競合がない。それにより、MAC層でのコンテンションやパケット衝突が少なく、再送タイムアウトもほとんど発生しないため、提案手法の効果がほとんど発揮されない結果となった。

4フローの場合、図2(b)より提案手法はNewVegasよりも低下してしまった。フロー数が増えると、他フローとの競合が発生するため、MAC層でのコンテンションやパケット衝突、再送タイムアウトが頻繁に起こる状況になっていく。この場合、提案手法はウィンドウサイズを元に戻すので、提案手法のウィンドウサイズが大きくなる。Chainトポロジ2Mbpsではウィンドウサイズが1に近いほど低遅延かつ高スループットの傾向にあり、提案手法はウィンドウサイズが若干大きくなる分、図3(b)のように遅延が大きくなりスループット低下の影響を受けてしまった。

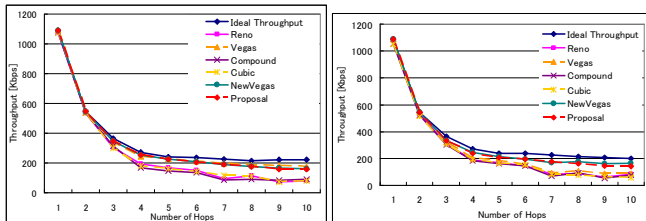


図2. 各方式のスループット (a)1flow (b)4flow

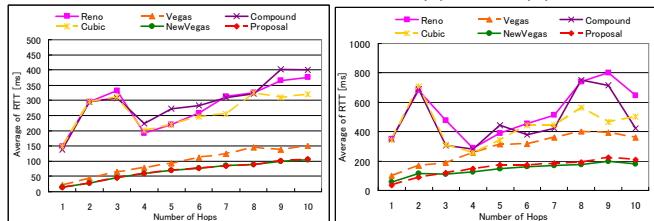


図3. 各方式の平均RTT (a)1flow (b)4flow

4.2 Random トポロジ

次に、より現実的な環境を想定して、ランダムノード配置とノード移動のあるRandomトポロジにて実験を行う。ノード数を20[node]、ノード移動速度を1~5[m/s]、シミュレーション範囲を500[m]×500[m]、フロー数を6[flow]とする。6フローの合計スループットと平均RTTをそれぞれ図4と図5に示す。

図4より、トータルスループットは、提案手法が一番高く、NewVegasに比べて約6.8%改善した。Random環境では、ノード移動もあるため再送タイムアウトが頻繁に起こる。加えてLDA-RTOで非輻輳と判断される割合が高いため、提案手法は再送タイムアウト時の状態に戻して再開するため、不必要なウィンドウ減少を防ぐことができ、帯域を有効に使えることができたためである。また、図5より、提案手法の平均RTTはNewVegasに比べて、約7.2%の増加に留まった。しかし、低遅延の特徴を活かすことができていると言える。

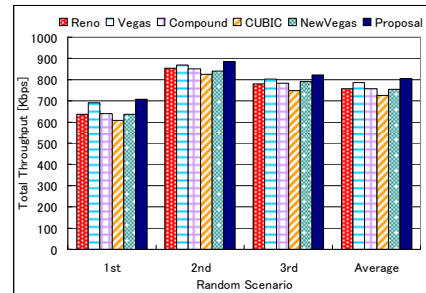


図4. トータルスループット

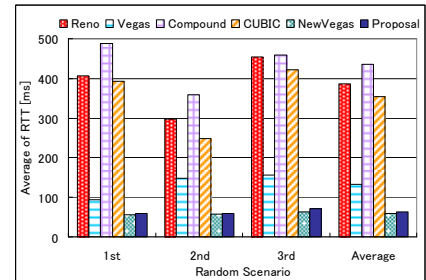


図5. 平均RTT

5 まとめ

本研究では、[2]で提案された輻輳制御方式のパケットロス後の対応を改良することで、ランダム環境においても、低遅延を維持しつつ、スループットを改善することができた。今後の課題として、TCPの理想スループット解析(帯域の上限)を利用した輻輳制御などが挙げられる。

6 参考文献

- [1] L.S.Brakmo, L.L.Perterson, "TCP Vegas: End-to-End Congestion Avoidance on a Global Internet", IEEE Journal on Selected Areas in Communication, Vol.13, August 1995.
- [2] 飯窪尚也, 甲藤二郎, "無線マルチホップアドホックネットワークにおける TCP-Vegas を拡張した輻輳制御方式", 信学技報 NS2009-174, Mar.2010.
- [3] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang: "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", In proc. of ACM Mobicom 2001.
- [4] M.Park, S.Chung, "Distinguishing the Cause of TCP Retransmission Timeouts in Multi-hop Wireless Networks", HPCC, 2010 12th IEEE International Conference on, Sep. 2010.